

Projet - Programmation Orientée Objet

Description

L'évaluation sommative de l'UA 2 sera un projet qui peut être fait en **groupe de 2 ou 3 élèves**.

Le projet doit consister d'une application avec interface Console qui démontre une bonne compréhension des sujets touchés durant le cours. Le sujet du projet est votre décision. (Si vous êtes à court d'inspiration, quelques idées seront listées plus bas.)

Vous aurez à remplir un formulaire sur votre choix de projet et il est possible que je refuse une idée de projet si plusieurs groupes ont déjà choisi le même concept parmi votre groupe de classe. Premiers arrivés, premiers servis! De plus, il est possible que j'impose que de la complexité supplémentaire soit ajoutée à votre projet si je juge qu'il est trop simple. Ceci sera ajusté en fonction du nombre de membres dans le groupe.

Vous aurez, à chaque semaine d'ici la fin du projet, la chance de recevoir une rétroaction sur votre travail à date durant la période de travail en classe. Vous aurez, comme à l'habitude des évaluations formatives chaque semaine, mais les périodes de travail en classe d'ici la fin du projet seront dédiées principalement au projet pour faciliter la coordination entre les horaires des étudiants et avoir accès à ma rétroaction et mon aide.

Votre projet doit répondre aux critères suivants:

1. Contenir au moins 3 classes, en plus d'une classe "maître" qui contrôlera l'application. Le fichier Program.cs (le point d'entrée) devrait initialiser un objet de cette "classe maître" pour démarrer l'exécution de l'application.
2. Contenir et faire usage d'un enum.
3. Faire une bonne utilisation des modificateurs d'accès (surtout *public* et *private*); s'assurer que seules les méthodes et les variables qui doivent nécessairement être visibles de l'extérieur de la classe en question le sont.
4. Utiliser les Propriétés (mutateurs/accesseurs - get/set) lorsque approprié
5. Au moins une des classes doit implémenter une surcharge d'opérateur et cet opérateur doit être utilisé dans le programme.
6. Faire utilisation de méthodes et de variables membres statiques (*static*) à au moins 2 occasions.
7. Inclure la documentation Visual Studio XML pour chaque méthode *public* de vos classes.
8. Être robuste (Si des entrées invalides sont entrées, le programme devrait être en mesure de ne pas planter et de gérer ces cas avec des boucles et/ou des exceptions)
9. Bonne utilisation des conventions d'écriture de code (classes et méthodes en *CassePascal*, noms de variables en *camelCasing*), noms de méthodes et variables descriptifs, etc.
- 10. Inclure un diagramme de classe.**
- 11. Inclure un document Word qui explique le principe de l'application et un bref guide d'utilisation incluant des captures d'écran.**

Une note sur l'intégrité scolaire

Il est possible que votre choix de projet comporte des aspects très complexes que vous ne pouvez pas résoudre vous même (exemple: Vous décidez de créer comme projet un jeu à deux joueurs qui est joué par un humain versus l'ordinateur; vous décidez de faire de la recherche sur des algorithmes à implémenter pour donner une "intelligence artificielle" au programme.) Vous pouvez faire recours à des ressources en ligne, **en autant que les critères suivants sont respectés:**

Si vous n'êtes pas certain à 100% si une ressource que vous utilisez répond à ces critères, **demandez-le moi afin de ne pas risquer d'obtenir une note de zéro.**

Comme d'habitude, tout le contenu du cours et la documentation de Microsoft sont des ressources que vous pouvez utiliser à volonté.

Présentation

Vous devrez, à la fin du projet, faire une présentation via Teams (environ 10 minutes) de votre programme. La présentation devrait commencer par une explication de votre projet, suivi de la présentation de l'architecture en démontrant le diagramme de classe. Ensuite, vous devrez faire un survol du code source de l'application et une démonstration de l'application en action.

Correction

Le projet vaut en totalité **35% de votre note finale**. Le poids des diverses parties du projet vont comme suit:

- Code source (Fonctionnalité, respect des critères, etc.): **70%**
- Documentation (Document Word, diagrammes): **15%**
- Présentation: **15%**

Tous les étudiants du groupe recevront la même note pour les deux premières parties. Les étudiants recevront une note individuelle pour la présentation en fonction de leur participation.

Dates limites

Le projet doit être soumis et présenté au plus tard à la **semaine 11** (18 au 22 Juillet 2022). Si certains groupes sont prêts à donner leur présentation lors de la **semaine 10** (11 au 15 Juillet 2022), il sera possible de le faire.

Une archive .zip devra être soumise contenant:

- La solution (Fichier .sln et le dossier de projet contenant le code source)
- Le document Word décrivant le projet
- Le diagramme de classes

La date limite pour la soumission de ce fichier est **48 heures avant votre cours durant la semaine 11**. Cette date limite sera reflétée dans le pigeonier de votre section de cours.

Avant de commencer

Veuillez remplir le formulaire à l'adresse suivante pour m'indiquer les membres de votre groupe, et une courte description de votre projet. (Seulement un membre de l'équipe a besoin de soumettre le formulaire). **Ne tardez pas, car votre projet pourrait être refusé.**

<https://forms.gle/uuY1V6xfr743wHUm6>

Note: Vous devez être connecté à la suite Google avec votre compte de la Cité. Vous pouvez vous en assurer en choisissant l'application G Suite à partir du menu "Apps" du portail.

Resources

Outil de diagramme de classe recommandé:

<https://online.visual-paradigm.com/diagrams/solutions/free-class-diagram-tool/>

Idées de projets

Si vous n'avez pas d'inspiration, sentez-vous libre de consulter la liste ci-dessous pour une idée potentielle de projet.

- Une calculatrice géométrique qui permet à l'utilisateur de créer des formes géométriques diverses, avec option de faire des calculs à partir de celles-ci (ex. Permettre à l'utilisateur de créer un cercle, un rectangle, ou un triangle et ensuite obtenir de l'information telle que son périmètre, sa surface, etc. ainsi que de comparer deux formes d'un même type pour savoir si elles sont équivalentes, ou si un cercle peut être inséré à l'intérieur d'un rectangle)
- Une application de magasinage qui permet à l'utilisateur de consulter une liste de produits disponibles, d'en ajouter à son panier, d'obtenir le total de son panier, et de créer une commande. L'utilisateur pourrait ensuite être en mesure de voir son historique de commandes.
- Une application de gestion de contacts qui permet à l'utilisateur de créer un contact, d'ajouter une ou plusieurs adresses courriels, adresses résidentielles, téléphone, etc.
- Une application de budget qui permet à l'utilisateur d'entrer son revenu, ses épargnes et ses dépenses, d'assigner des pourcentages de son revenu à certaines catégories, et d'obtenir des statistiques sur sa situation financière.
- Un jeu, comme par exemple:
 - *Connect 4*
 - Bonhomme pendu (*hangman*)
 - Dammes (*checkers*)
 - *Blackjack*

Une note pour les étudiants enthousiastes

Si vous avez une idée de projet plus ambitieux mais que vous avez besoin de support, il me fera grand plaisir de vous offrir des conseils. Le projet est une évaluation sommative, donc bien sûr il sera à vous de faire le travail, mais je pourrai vous diriger vers des ressources ou vous présenter des concepts qui sont en dehors de la portée du cours qui pourront vous être utiles!